doi.org/10.54473/IJTRET.2024.8507
International Journal of Trendy Research in Engineering and Technology
Volume 8 Issue 5 October 2024
ISSN No. 2582-0958

ADVANCED TECHNIQUE FOR REQUIREMENT ANALYSIS OF SOFTWARE PROJECTS

Syed Osama Ali Shah
Bahria University, Karachi
s.osamaali72@gmail.com

Received 02 July 2024 Received in revised form 10 July 2024 Accepted 11 July 2024
Available online 16 July 2024

ABSTRACT

Following the requirements elicitation activity, the team will then carry out the requirements analysis to identify the requirements that have the potential to cause issues in the future. There are several different approaches to requirement analysis that can be used to determine the specific requirements that may either compete with or overlap with those of other requirements. We can perform requirement analysis in a better and more efficient manner with the help of the mode proposed in this paper. Requirement analysis is an important activity after requirement analysis and a proper mechanism is required to make this activity more precise and efficient. In this paper, we present a model that can be used to perform an analysis of requirements and highlight those that overlap with or are in direct opposition to the requirements of another requirement. Any requirements that are incompatible with one another or that overlap will be called out. Following the calculation of the overlapping and conflicting fitness of each of the requirements. we will be able to identify all the requirements that have a higher rate of overlapping or conflicting.

Keywords: Requirement Analysis, Software Engineering, Conflicting Fitness, Overlapping Fitness. Requirement Elicitation.

I.INTRODUCTION

The software engineering process includes a wide variety of important activities, one of the most important of which is requirements analysis. Without conducting a thorough requirement analysis first, it is impossible to develop high-quality software. The term "requirement analysis" refers to the process of identifying requirements that overlap with one another, requirements that are in direct opposition to one another, and requirements that cannot be implemented due to financial or logistical constraints. During the requirements analysis phase, we also bring to light requirements that cannot be satisfied realistically. Following the requirements elicitation activity, the team will then carry out the requirements analysis to identify the requirements that have the potential to cause issues in the future.

There are several different approaches to requirement analysis that can be used to determine the specific requirements that may either compete with or overlap with those of other requirements. The project can be jeopardized if some of the requirements overlap with those of others or if they conflict with those of other requirements. If we implement a requirement that conflicts with a few other requirements, then it will most certainly affect the functionality of the software project that is currently being developed. When people use the software that we developed, there is a possibility that they will encounter bugs. Requirements might contradict one another and are typically negotiable. Key stakeholders should assess requirements, all requirements should be prioritized, and requirements that clash should be reevaluated. To make the conclusion accessible to each member of the project team, the reviewed and authorized priority must be documented [1].

To establish how acceptable each user group will find the new structure, a benefit analysis can be done. In contrast to single-user, single-task products, systems that affect entire work processes are best suited for the use of task allocation charts and cost-benefit analyses. To create a more palatable solution for all parties, they also give the chance to reconsider the system architecture or user responsibilities [2].

II.REQUIREMENT MANAGEMENT & ITS IMPORTANCE

Inconsistencies between requirements, project plans, and work deliverables are addressed through requirement management. Information is gathered, needs are defined, and requirements analysis and documentation are all part of requirement engineering [3]. One of the key components of requirement management is traceability, which refers to the capacity to describe and track the evolution of demand and its relationships with other things in both directions

(forward and backward) [4]. Additionally, as requirements could still change after development and throughout maintenance, requirements management is a continual process [5,6]

In this paper, we present a model that can be used to perform an analysis of requirements and highlight those that either overlap with or are in direct opposition to the requirements of another requirement. All of the requirements will be listed in a table, and any requirements that are incompatible with one another or that overlap will be called out. Following the completion of the calculation of the overlapping and conflicting fitness of each of the requirements, we will be able to identify all of the requirements that have a higher rate of either overlapping or conflicting. We will be able to construct software that is of a higher quality and more stable if we use this model.

Model:

Conflicting requirements are those kinds of requirements that are inconsistent with a small number of other requirements, and this can lead to the production of software that might not function correctly. This is because the implementation of two conflicting requirements and their inconsistency led to the problem. On the other side, we have overlapping needs, which may be rather common. Because of this, there are consistency rules among those requirements, and if those rules are broken, we might reach a stage where those requirements are inconsistent. We are going to make use of an innovative method for requirement analysis that will assist us in locating overlapping and conflicting requirements. In addition to this, the method will allow us to compute the fitness of each demand on an individual basis.

Our model can be utilized to sketch out the criteria that compete with one another and overlap. We are going to emphasize the criteria with the use of numerical indicators, which will help classify the requirements into three distinct categories. The first indicator will highlight those requirements that do not overlap or conflict with a particular requirement, the second indicator will highlight those requirements that are incompatible with that requirement, and finally, our third indicator will highlight those requirements that overlap with the particular requirement.

0 = Neither Overlapping nor Conflicting
1 = Overlapping
2 = Conflicting

In a table, we will first list all the requirements, and then, in front of that table, we will list all the requirements that are incompatible with one another and that overlap. In the column and row, we will list all the requirements and give them their respective names. R1, R2 , R3 etc. In the column of the requirement that overlaps with the requirement, we will write "one," and in the column of the requirement that conflicts with the particular requirement, we will write "two." We shall write the number zero in the column of the requirement that does not overlap nor is it in conflict with the particular requirement.

On this basis, we will be able to compile a list of all the criteria that, in one way or another, compete with, duplicate, or otherwise overlap with another requirement. We will be able to identify all the requirements that were obtained from the client that did not conflict with one another and did not overlap thanks to this activity, which will help us prioritize the requirements that were given to us.

	R1	R2	R3	R4
R1	.	0	2	1
R2	2	.	2	0
R3	1	0	.	0
R4	2	2	1	.

Fig 1. Requirement Score Table

Fig 1. displays a requirement score table with a mapped value of conflicting and overlapping between the requirements

1. Overlapping Fitness:

Overlapping fitness will be the first one we will discuss in this paper. Overlapping is one of the couple of important points that we have to analyze about each of the requirements. If a requirement doesn't overlap with any other requirement, then we can be assured that our requirements won't have inconsistent behavior at any stage.

We are going to place the total number of requirements that overlap with that particular requirement in the numerator, and below the fraction bar, we are going to place total requirements and subtract it by one. This is the new formula that we are going to use to calculate the overlapping fitness of requirements. The solution will be our overlapping fitness for that particular requirement. It is vital to reduce the total requirement by one since inside the total requirements we also have that one individual requirement for which we are calculating the fitness. Because a requirement cannot overlap or be in direct opposition with itself, we have to take one away from the total number of requirements to prevent any kinds of irregularities.

$$\text{Overalapping Fitness} = \frac{\text{Overlapping Requirements}}{\left[\text{Total Requirements} - 1\right]}$$

Fig.2 Formula Overlapping Fitness

Fig .2 is explaining the formula for calculating overlapping fitness of the requirements of the software project

2. Conflicting Fitness:

Another important aspect of the requirements that need to be checked off is whether or not they conflict with one another. It is yet another significant aspect that needs to be evaluated so that more accurate analyses of requirements can be performed. We also need to ensure that our requirements are consistent and that they won't give rise to any kind of complications in the future. If conflicting needs are not resolved appropriately, the program may behave abnormally, and in extreme circumstances, the system will fail. We are going to make a few tweaks to the formula that we used for our overlapping requirement fitness to adapt it so that it can also be utilized for our conflicting requirement fitness. Below the fraction line, our denominator will remain the same, and we will continue to take the total number of requirements and reduce it by one. The numerator will contain all of the requirements that are incompatible with the specific requirement being considered. Following the completion of the calculation of the overlapping and conflicting fitness of the demand, we will construct a table for the analyses.

$$\text{Conflicting Fitness} = \frac{\text{Conflicting Requirements}}{\left[\text{Total Requirements} - 1\right]}$$

Fig. 3. Formula Conflicting Fitness

Fig. 3 explains the formula for calculating conflicting fitness for the requirements of the software project

3. Requirements Table:

Following the completion of the computations necessary to determine whether or not all of the requirements overlap or conflict with one another, the next step is to construct a table that will contain all of the requirements, along with an indication of whether or not they overlap or conflict with one another. This table may be used to examine all of the needs based on their suitability, and decisions can be made to either keep a requirement or delete it. This will allow us to identify all of the requirements that have the potential to create obstacles in the process of completing our project.

We can make use of this table while negotiating with customers, and we are also able to display the results of our analysis to customers.

It will assist them in making a more informed choice regarding which standards should be maintained and which should be eliminated, the primary objective is the production of a flawless and effective product that can be utilized to assist in the resolution of issues. This document needs to be presented to the customer for us to receive authorization from the client to delete any of the requirements, so we must do so. The analyst and the team need to persuade the customer to get rid of the needs that would result in complications in the future.

Requirement Name	Overlapping Fitness	Conflicting Fitness
R1	72.34%	82.78%
R2	61.23%	08.21%
R3	51.98%	21.01%

Fig. 4 Requirement Fitness Score

Fig. 4 explains to us the fitness score of the individual requirements by showcasing their overlapping and conflicting fitness scores.

III. FUTURE WORK

Because the labor that is done now is intelligently automated. We are able to develop an AI-based smart system that, using the method proposed in this paper, can detect overlapping and conflicting requirements, calculate the fitness of overlapping and conflicting requirements, and then generate a report regarding the fit and unfit requirements based on overlap and conflict. They will be able to carefully assess and analyze the user requirements, and they will be able to provide their opinion on where the requirements are conflicting or overlapping, or where some of them are unable to implement due to some other reasons. This software has the potential to be a huge asset to the software company because it will allow them to do these things.

This piece of software could be constructed as a language model similar to ChatGPT or ChatSonic, for example. We can construct our very own language model, which will interact with the clients, collect all of the requirements from the clients, carry out the requirements analyses and check for overlapping and conflicting requirements, as well as check for requirements that cannot be implemented due to any constraints or perhaps are imaginary, among other things. In addition to this, it needs to be able to bargain with the customer and close the deal. The deployment of this technology will result in significant cost savings because it will take the position of business analysts. The job of a software company will be automated, and it will be possible for software development companies to save a significant amount of money as a result.

IV.CONCLUSION

We talked about a fresh and original method for analyzing the user needs that were gathered and calculating the overlapping and conflicting fitness of each of the user requirements that were gathered. It will be of great use to us because it will enable us to perform a more thorough analysis of the requirements and to visualize our data in a way that is both conventional and original. The analyst will be able to perform a better and more accurate analysis of the requirements with the assistance of this method and will be in a better position to share the results with the customer in a manner that is more appropriate during the negotiating meeting. The procedure under consideration Within the framework of this paradigm, it is possible to develop several intelligent AI-based tools that are capable of automating work, evaluating requirements, and negotiating with customers.

Further research can be done on this subject as requirement analyses are one of the most important parts of software development and many products, even in this day and age, are not completed due to poor requirement elicitation and analyses. In the future, we might see improved and advanced versions of the simple formula that was derived above. In the meantime, however, we can conduct more research on this subject.

REFERENCES

[1] S. T. Demirel and R. Das, “Software requirement analysis: Research challenges and technical approaches,” 2018 6th International Symposium on Digital Forensic and Security (ISDFS), Mar. 2018

[2] M. Maguire and N. Bevan, “User Requirements Analysis,” IFIP Advances in Information and Communication Technology, pp. 133–148, 2002, doi: 10.1007/978-0-387-35610-5_9.

[3] D. Pandey, U. Suman, A. K. Ramani, “Impact of Requirement Engineering Practices in Software Development Processes for Designing Quality Software Products”, National Conference on NCAFIS, DAVV, Indore, 2008.

[4] S. Lauesen, Software Requirements: Styles and Techniques, AddisonWesley,2002

[5] S. Blanchard & W. J. Fabrycky, Systems Engineering and Analysis, Prentice-Hall, 1981.

[6] R. Stevens, P. Brook, K. Jackson & S. Arnold, Systems Engineering - Coping with Complexity, Prentice Hall, London, 1998.